

DATA SUPPLYING DEVICE FOR PROCESSOR

Patent Number: JP11212788
Publication date: 1999-08-06
Inventor(s): SATOU HISATOMO
Applicant(s): TOSHIBA CORP
Requested Patent: ☐ JP11212788
Application Number: JP19980016159 19980128
Priority Number(s):
IPC Classification: G06F9/38
EC Classification:
Equivalents:

Abstract

PROBLEM TO BE SOLVED: To predictively perform data acquisition of a load instruction to speculatively carry out subsequent instructions, to reduce a data dependence relation period of the load instruction and a store instruction and to improve processor performance by supplying data acquired by referring to a specific table.

SOLUTION: A load index store table(LIST) 207 is accessed by using a load instruction address that is shown by a program counter(PC) 600 on an IF stage. A store index value table(SIVT) 307 is accessed by using a store instruction address that is obtained from the LIST 207. Data that is acquired from the SIVT 307 is supplied to succeeding instructions as an operand and the succeeding instructions are speculatively carried out. A comparator 500 compares data that is acquired from data memory 800 with the data acquired from the SIVT 307 on an MA stage of the load instruction. If both coincide, speculative execution is a success.

Data supplied from the esp@cenet database - 12

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-212788

(43) 公開日 平成11年(1999) 8月6日

(51) Int.Cl.⁵

G 0 6 F 9/38

識別記号

3 5 0

F I

G 0 6 F 9/38

3 5 0 B

審査請求 未請求 請求項の数 9 O L (全 13 頁)

(21) 出願番号

特願平10-16159

(22) 出願日

平成10年(1998) 1月28日

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 佐藤 寿倫

神奈川県川崎市幸区堀川町580番1号 株

式会社東芝半導体システム技術センター内

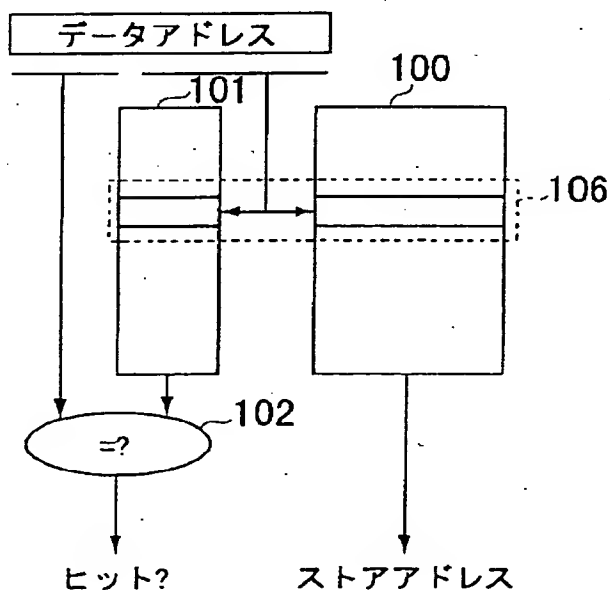
(74) 代理人 弁理士 三好 秀和 (外3名)

(54) 【発明の名称】 プロセッサのデータ供給装置

(57) 【要約】

【課題】 この発明は、ロード命令のデータ取得を予測的に行って後続命令を投機的に実行し、さらにロード命令とストア命令のデータ依存関係期間を短縮し、プロセッサの性能を向上させることを課題とする。

【解決手段】 この発明は、データのアドレスでインデックスされてストア命令の命令アドレスを保持する D I S T、ロード命令の命令アドレスでインデックスされてストア命令の命令アドレスを保持する L I S T ならびにストア命令の命令アドレスでインデックスされてデータの値を保持する S I V T を関連づけて機能させることによりロード命令の命令アドレスからデータ値を予測するように構成される。



【特許請求の範囲】

【請求項1】 プロセッサがデータメモリにアクセスしたデータをプロセッサに供給する装置であって、データのアドレスで参照されてデータのアドレスに対応して登録されたストア命令のアドレスを供給し、過去のストア命令の実行履歴に基づいて登録内容が登録更新されるデータインデックスストアテーブル (DIST) と、

ロード命令のアドレスで参照されてロード命令のアドレスに対応して登録されたストア命令のアドレスを供給し、過去のロード命令の実行履歴ならびに前記DISTに登録されたストア命令のアドレスに基づいて登録内容が登録更新されるロードインデックスストアテーブル (LIST) と、

ストア命令のアドレスで参照されてストア命令のアドレスに対応して登録されたデータを供給し、過去のストア命令の実行履歴に基づいて登録内容が登録更新されるストアインデックスバリュートーブル (SIVT) を備え、

前記DISTは同じデータのアドレスを参照するロード命令とストア命令を関係づけ、ロード命令のアドレスにしたがって前記LISTから得られたストア命令のアドレスで前記SIVTを参照して得られたデータを、ロード命令によりアクセスが予測されるデータとしてプロセッサに供給してなることを特徴とするプロセッサのデータ供給装置。

【請求項2】 プロセッサがデータメモリにアクセスしたデータをプロセッサに供給する装置であって、データのアドレスで参照されてデータのアドレスに対応して登録されたストア命令のアドレスを供給し、過去のストア命令の実行履歴に基づいて登録内容が登録更新されるデータインデックスストアテーブル (DIST) と、

ロード命令のアドレスで参照されてロード命令のアドレスに対応して登録されたストア命令のアドレスを供給し、過去のロード命令の実行履歴ならびに前記DISTに登録されたストア命令のアドレスに基づいて登録内容が登録更新されるロードインデックスストアテーブル (LIST) と、

ストア命令のアドレスで参照されてストア命令のアドレスに対応して登録されたデータを供給し、過去のストア命令の実行履歴に基づいて登録内容が登録更新されるストアインデックスバリュートーブル (SIVT) と、プロセッサがデータ供給装置を参照する以前のプロセッサの状態を保持する状態保持装置を備え、

前記DISTは同じデータのアドレスを参照するロード命令とストア命令を関係づけ、ロード命令のアドレスにしたがって前記LISTから得られたストア命令のアドレスで前記SIVTを参照して得られたデータを、ロード命令によりアクセスが予測されるデータとしてプロセ

ッサに供給し、プロセッサが前記データメモリにアクセスしてデータを獲得した際に、前記データ供給装置から得られた予測データと前記データメモリから得られたデータとを比較し、両者が不一致の場合には前記状態保持装置に保持された状態にプロセッサを復帰させることを特徴とするプロセッサのデータ供給装置。

【請求項3】 プロセッサがデータメモリにアクセスしたデータをプロセッサに供給する装置であって、データのアドレスで参照されてデータのアドレスに対応して登録されたストア命令のアドレスを供給し、過去のストア命令の実行履歴に基づいて登録内容が登録更新されるデータインデックスストアテーブル (DIST) と、

ロード命令のアドレスで参照されてロード命令のアドレスに対応して登録されたストア命令のアドレスを供給し、過去のロード命令の実行履歴ならびに前記DISTに登録されたストア命令のアドレスに基づいて登録内容が登録更新されるロードインデックスストアテーブル (LIST) と、

ストア命令のアドレスで参照されてストア命令のアドレスに対応して登録されたデータを供給し、過去のストア命令の実行履歴に基づいて登録内容が登録更新されるストアインデックスバリュートーブル (SIVT) と、プロセッサがデータ供給装置を参照する以前のプロセッサの状態を保持するリオーダバッファを備え、

前記DISTはデータのアドレスで参照されるエントリがデータのアドレスの一致を判別するタグフィールドとストア命令のアドレスを保持するストア命令アドレスフィールドと保持されているストア命令のアドレスが有効か否かを示すバリッドビットからなり、前記LISTはロード命令のアドレスで参照されるエントリがロード命令のアドレスの一致を判別するタグフィールドとストア命令のアドレスを保持するストア命令アドレスフィールドと保持されているストア命令のアドレスが有効か否かを示すバリッドビットからなり、前記SIVTはストア命令のアドレスで参照されるエントリがストア命令のアドレスの一致を判別するストア命令アドレスフィールドとデータを保持するデータフィールドと保持されているデータが有効か否かを示すバリッドビットからなり、

前記DISTは同じデータのアドレスを参照するロード命令とストア命令を関係づけ、ロード命令のアドレスにしたがって前記LISTから得られたストア命令のアドレスで前記SIVTを参照して得られたデータを、ロード命令によりアクセスが予測されるデータとしてプロセッサに供給し、プロセッサが前記データメモリにアクセスしてデータを獲得した際に、前記データ供給装置から得られた予測データと前記データメモリから得られたデータとを比較し、両者が不一致の場合には前記リオーダバッファに保持された状態にプロセッサを復帰させることを特徴とするプロセッサのデータ供給装置。

【請求項4】 前記DISTのバリッドビットは、登録されているデータをバイト単位で有効か否かを示すことを特徴とする請求項3記載のプロセッサのデータ供給装置。

【請求項5】 前記LISTのエントリには、データあるいはデータのアドレスの比較時に一致した場合は増加され不一致の場合には減少するカウンタ値が保持されるカウンタフィールドが設けられ、前記LISTの参照時に前記カウンタフィールドのカウンタ値に基づいて、データ供給装置から獲得されたデータあるいはデータアドレスを用いて後続の命令を投機的に実行するか否かを決定することを特徴とする請求項3又は4記載のプロセッサのデータ供給装置。

【請求項6】 前記LISTは、ストア命令のアドレスに加えてデータのアドレスを登録して供給することを特徴とする請求項1、2、3、4又は5記載のプロセッサのデータ供給装置。

【請求項7】 前記SIVTは、ストア命令のアドレスに対応したデータに加えてデータのアドレスを供給することを特徴とする請求項1、2、3、4、5又は6記載のプロセッサのデータ供給装置。

【請求項8】 プロセッサがデータのアドレスを計算した際に、前記LIST又は前記SIVTから得られたデータのアドレスと計算して得られたデータのアドレスとを比較し、両者が不一致の場合には前記状態保持装置又は前記リオーダバッファに保持された状態にプロセッサを復帰させることを特徴とする請求項6又は7記載のプロセッサのデータ供給装置。

【請求項9】 前記LISTを参照するパイプライン処理のステージと前記SIVTを参照するパイプライン処理のステージが異なるステージであることを特徴とする請求項1、2、3、4、5、6、7又は8記載のプロセッサのデータ供給装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、ロード命令アドレスから読み出すべきデータをプロセッサに供給するプロセッサのデータ供給装置に関する。

【0002】

【従来の技術】 まず最初に、一度に一命令しか実行できない単純なスカラプロセッサを用いてロード命令の実行方法を説明し、単一のロード命令実行時の問題を説明する。続いて、命令の実行時に命令の並べ換えが可能なアウトオブオーダー実行型のプロセッサを用いて、同時にストア命令とロード命令を連続実行する時の問題を説明する。

【0003】 図20は標準的な5段のRISC型プロセッサのパイプラインを表している。これは、文献「コンピュータ・アーキテクチャ」（ヘネシー他著、日経BP社）等にも記載されており、非常に基本的なプロセッサ

のパイプラインである。このパイプライン処理では、ある一つの演算は5つのステージによって実行される。それらのステージは、命令フェッチ（IF）ステージ、命令デコード（ID）ステージ、演算（EX）ステージ、メモリアクセス（MA）ステージ、そして書き戻し（WB）ステージである。IFステージでは命令メモリから命令がフェッチされる。IDステージでは命令が翻訳され、同時にレジスタファイルにアクセスし演算に必要なオペランドが獲得される。EXステージでは演算が実行される。データメモリにアクセスする命令の場合には、EXステージでデータアドレスが計算される。MAステージではデータメモリにアクセスする。EXステージでは計算されたアドレスを用いてデータを獲得する。WBステージでは演算結果とデータメモリから読み出したデータをレジスタファイルに書き戻す。

【0004】 図21に命令が連続する場合のパイプライン処理の振舞いを示す。先行する命令が標準的な算術演算命令（図21では加算（add）命令）である場合には、後続の命令を連続して実行することが可能である。図21中の矢印は演算結果のバイパスを表している。しかし、先行する命令がロード命令の場合には状況が異なる。ロード命令はMAステージを終了しなければデータを獲得できないので、直後の命令はEXステージの開始までに演算に必要なデータを獲得できない。このため、後続の命令はロード命令の実行が終了するまでEXステージの実行を待機する必要がある。すなわち、ロード命令はデータアドレス計算とメモリアクセスという2つの操作を行なうため、ロード命令の結果を用いる場合は、他の演算結果を用いる場合と比べてデータの依存関係が存在する期間が長く、プロセッサの性能向上の妨げとなっていた。

【0005】 次に、図22に示す命令列を用いて、アウトオブオーダー実行型のプロセッサで命令を実行する場合の問題を説明する。

【0006】 先行するストア（sw）命令のアドレスを計算するためのレジスタr2は確定しておらず、一方ロード（lw）命令のアドレスを実行するためのレジスタr3は確定しているものとする。加算（add）命令のオペランドr20、r21も確定しているものとする。sw命令はオペランドが揃っていないため実行を待機している。add命令はオペランドが揃っているためsw命令を追い越して実行を開始することが可能である。lw命令もオペランドが揃っているため実行を開始できるように思えるが、sw命令との間の依存性が解消されていないので実行を開始できない。つまり、先行するストア命令がデータを保存するデータアドレスがわからなければ、後続のロード命令は実行できない。なぜなら、もし、ストア命令の計算するデータアドレスとロード命令のデータアドレスとが一致する場合は、ストア命令が保存しようとしているデータをロード命令が読み出さなけ

ればならないからである。このため、オペランドが揃っていてもロード命令は待機中のストア命令を追い越して実行できないので、命令の実行待機時間が長くなりプロセッサの性能を向上することはできない。

【0007】

【発明が解決しようとする課題】以上説明したように、従来のプロセッサにおいてロード命令を実行する場合に、アドレス計算とメモリアクセスという2つの操作が行われるため、他の命令に比べて依存関係のパスが長くなり、さらに、先行するストア命令が実行されないと後続のロード命令は実行できないという制約があった。これらの制約は命令レベル並列度の抽出の妨げとなり、プロセッサの実行性能を低下させてしまうという不具合を招いていた。

【0008】そこで、この発明は、上記に鑑みてなされたものであり、その目的とするところは、ロード命令のデータ取得を予測的に行って後続命令を投機的に実行し、さらにロード命令とストア命令のデータ依存関係期間を短縮し、プロセッサの性能を向上させるプロセッサのデータ供給装置を提供することにある。

【0009】

【課題を解決するための手段】上記目的を達成するために、請求項1記載の発明は、プロセッサがデータメモリにアクセスしたデータをプロセッサに供給する装置であって、データのアドレスで参照されてデータのアドレスに対応して登録されたストア命令のアドレスを供給し、過去のストア命令の実行履歴に基づいて登録内容が登録更新されるデータインデックスストアテーブル(DIST)と、ロード命令のアドレスで参照されてロード命令のアドレスに対応して登録されたストア命令のアドレスを供給し、過去のロード命令の実行履歴ならびに前記DISTに登録されたストア命令のアドレスに基づいて登録内容が登録更新されるロードインデックスストアテーブル(LIST)と、ストア命令のアドレスで参照されてストア命令のアドレスに対応して登録されたデータを供給し、過去のストア命令の実行履歴に基づいて登録内容が登録更新されるストアインデックスバリュートーブル(SIVT)を備え、前記DISTは同じデータのアドレスを参照するロード命令とストア命令を関係づけ、ロード命令のアドレスにしたがって前記LISTから得られたストア命令のアドレスで前記SIVTを参照して得られたデータを、ロード命令によりアクセスが予測されるデータとしてプロセッサに供給してなることを特徴とする。

【0010】請求項2記載の発明は、プロセッサがデータメモリにアクセスしたデータをプロセッサに供給する装置であって、データのアドレスで参照されてデータのアドレスに対応して登録されたストア命令のアドレスを供給し、過去のストア命令の実行履歴に基づいて登録内容が登録更新されるデータインデックスストアテーブル

(DIST)と、ロード命令のアドレスで参照されてロード命令のアドレスに対応して登録されたストア命令のアドレスを供給し、過去のロード命令の実行履歴ならびに前記DISTに登録されたストア命令のアドレスに基づいて登録内容が登録更新されるロードインデックスストアテーブル(LIST)と、ストア命令のアドレスで参照されてストア命令のアドレスに対応して登録されたデータを供給し、過去のストア命令の実行履歴に基づいて登録内容が登録更新されるストアインデックスバリュートーブル(SIVT)と、プロセッサがデータ供給装置を参照する以前のプロセッサの状態を保持する状態保持装置を備え、前記DISTは同じデータのアドレスを参照するロード命令とストア命令を関係づけ、ロード命令のアドレスにしたがって前記LISTから得られたストア命令のアドレスで前記SIVTを参照して得られたデータを、ロード命令によりアクセスが予測されるデータとしてプロセッサに供給し、プロセッサが前記データメモリにアクセスしてデータを獲得した際に、前記データ供給装置から得られた予測データと前記データメモリから得られたデータとを比較し、両者が不一致の場合には前記状態保持装置に保持された状態にプロセッサを復帰させることを特徴とする。

【0011】請求項3記載の発明は、プロセッサがデータメモリにアクセスしたデータをプロセッサに供給する装置であって、データのアドレスで参照されてデータのアドレスに対応して登録されたストア命令のアドレスを供給し、過去のストア命令の実行履歴に基づいて登録内容が登録更新されるデータインデックスストアテーブル(DIST)と、ロード命令のアドレスで参照されてロード命令のアドレスに対応して登録されたストア命令のアドレスを供給し、過去のロード命令の実行履歴ならびに前記DISTに登録されたストア命令のアドレスに基づいて登録内容が登録更新されるロードインデックスストアテーブル(LIST)と、ストア命令のアドレスで参照されてストア命令のアドレスに対応して登録されたデータを供給し、過去のストア命令の実行履歴に基づいて登録内容が登録更新されるストアインデックスバリュートーブル(SIVT)と、プロセッサがデータ供給装置を参照する以前のプロセッサの状態を保持するリオーダバッファを備え、前記DISTはデータのアドレスで参照されるエントリがデータのアドレスの一致を判別するタグフィールドとストア命令のアドレスを保持するストア命令アドレスフィールドと保持されているストア命令のアドレスが有効か否かを示すバリッドビットからなり、前記LISTはロード命令のアドレスで参照されるエントリがロード命令のアドレスの一致を判別するタグフィールドとストア命令のアドレスを保持するストア命令アドレスフィールドと保持されているストア命令のアドレスが有効か否かを示すバリッドビットからなり、前記SIVTはストア命令のアドレスで参照されるエン

リがストア命令のアドレスの一致を判別するストア命令アドレスフィールドとデータを保持するデータフィールドと保持されているデータが有効か否かを示すバリッドビットからなり、前記DISTは同じデータのアドレスを参照するロード命令とストア命令を関係づけ、ロード命令のアドレスにしたがって前記LISTから得られたストア命令のアドレスで前記SIVTを参照して得られたデータを、ロード命令によりアクセスが予測されるデータとしてプロセッサに供給し、プロセッサが前記データメモリをアクセスしてデータを獲得した際に、前記データ供給装置から得られた予測データと前記データメモリから得られたデータとを比較し、両者が不一致の場合には前記リオーダバッファに保持された状態にプロセッサを復帰させることを特徴とする。

【0012】請求項4記載の発明は、請求項3記載のプロセッサのデータ供給装置において、前記DISTのバリッドビットは、登録されているデータをバイト単位で有効か否かを示すことを特徴とする。

【0013】請求項5記載の発明は、請求項3又は4記載のプロセッサのデータ供給装置において、前記LISTのエンタリには、データあるいはデータのアドレスの比較時に一致した場合は増加され不一致の場合には減少するカウンタ値が保持されるカウンタフィールドが設けられ、前記LISTの参照時に前記カウンタフィールドのカウント値に基づいて、データ供給装置から獲得されたデータあるいはデータアドレスを用いて後続の命令を投機的に実行するか否かを決定することを特徴とする。

【0014】請求項6記載の発明は、請求項1、2、3、4又は5記載のプロセッサのデータ供給装置において、前記LISTは、ストア命令のアドレスに加えてデータのアドレスを登録して供給することを特徴とする。

【0015】請求項7記載の発明は、請求項1、2、3、4、5又は6記載のプロセッサのデータ供給装置において、前記SIVTは、ストア命令のアドレスに対応したデータに加えてデータのアドレスを供給することを特徴とする。

【0016】請求項8記載の発明は、請求項6又は7記載のプロセッサのデータ供給装置において、プロセッサがデータのアドレスを計算した際に、前記LIST又は前記SIVTから得られたデータのアドレスと計算して得られたデータのアドレスとを比較し、両者が不一致の場合には前記状態保持装置又は前記リオーダバッファに保持された状態にプロセッサを復帰させることを特徴とする。

【0017】請求項9記載の発明は、請求項1、2、3、4、5、6、7又は8記載のプロセッサのデータ供給装置前記LISTを参照するパイプライン処理のステージと前記SIVTを参照するパイプライン処理のステージが異なるステージであることを特徴とする。

【0018】

【発明の実施の形態】以下、図面を用いてこの発明の実施形態を説明する。

【0019】この発明は、ロード命令がメモリから読み出す値をデータアドレス計算が終了する以前にそれまで蓄積された過去の履歴を参照して予測し、この予測された値を用いて後続の命令を投機的に実行するものである。データアドレス計算に要する時間を迂回するために、データアドレスをストア命令の命令アドレスに名前を付け換え、さらに同じデータアドレスにアクセスするロード命令とストア命令を関連づけ、ロード命令の命令アドレスからストア命令の命令アドレスへの変換を可能にする。以上の操作により、ロード命令アドレスからストア命令アドレス、ストア命令アドレスからデータアドレスへの変換が可能になり、データアドレスを計算せずとも、ロード命令アドレスから読み出すべきデータを参照することができるようになる。

【0020】請求項1、2又は3記載の発明の一実施形態に係るプロセッサのデータ供給装置は、例えば標準的な構成のプロセッサに対してデータを供給するものであり、以下に示す3つのテーブルを備えている。この3つのテーブルは、データアドレスでインデックスされてストア命令アドレスを保持するデータインデックスストアテーブル(DIST)と、ロード命令アドレスでインデックスされてストア命令アドレスを保持するロードインデックスストアテーブル(LIST)と、ストア命令アドレスでインデックスされてデータの値を保持するストアインデックスバリュートーブル(SIVT)である。各テーブルはキャッシュメモリと同様に構成されており、各テーブルの連想度には制約はないものとする。なお、ここでは、各テーブルはダイレクトマップ(連想度1)で構成されているものとして説明する。

【0021】図1はDISTの動作と構成を表している。図1において、DISTはタグアレイ101とデータアレイ100、比較器102とから構成され、データアドレスであるエンタリ106がセレクトされる。タグアレイ101から得られるタグとデータアドレスの上位ビットを比較し、一致すればデータアレイ100から得られたストア命令アドレスは有効である。

【0022】図2はDISTのエンタリ106を表している。図2において、各エンタリ106はタグフィールド103、ストア命令のアドレスフィールド104、バリッドビット105からなる。DISTはデータアドレスのうちエンタリ数分に相当する下位ビットでセレクトされる。タグフィールド103には残りの上位ビットが保存されている。セレクトされたエンタリ106に保存されているタグとデータアドレスの上位ビットを比較し、一致すればそのデータアドレスに対応するストア命令アドレスが保持されていることになる。ストア命令アドレスフィールド104には、インデックスとして用いられたデータアドレスに以前データをストアした命令の

アドレスが保持されている。バリッドビット105はそのエントリ106が有効であるかどうかを表している。プロセッサの起動時にはバリッドビット105は全て無効を表しており、DISTの各エントリ106にデータが保存されるとバリッドビット105が有効になる。

【0023】図3はLISTの動作と構成を表している。図3において、LISTはタグアレイ201とデータアレイ200、比較器202とから構成され、ロード命令アドレスであるエントリ206がセレクトされる。タグアレイ201から得られるタグとロード命令アドレスの上位ビットを比較し、一致すればデータアレイ200から得られたストア命令アドレスは有効である。

【0024】図4はLISTのエントリ206を表している。図4において、各エントリ206はタグフィールド203、ストア命令アドレスフィールド204、バリッドビット205からなる。LISTはロード命令アドレスのうちエントリ数分に相当する下位ビットでセレクトされる。タグフィールド203には残りの上位ビットが保存されている。セレクトされたエントリ206に保存されているタグとロード命令アドレスの上位ビットを比較し、一致すればそのロード命令アドレスに対応するストア命令アドレスが保持されていることになる。ストア命令アドレスフィールド204には、該ロード命令が参照しようとしているデータを保存したストア命令のアドレスが保存されている。バリッドビット205はそのエントリ206が有効であるかどうかを表している。プロセッサの起動時にはバリッドビット205は全て無効を表しており、LISTの各エントリ206にデータが保存されるとバリッドビット205が有効になる。

【0025】図5はSIVTの動作と構成を表している。図5において、SIVTはタグアレイ301とデータアレイ300、比較器302とから構成され、ストア命令アドレスであるエントリ306がセレクトされる。タグアレイから得られるタグとストア命令アドレスの上位ビットとを比較し、一致すればデータアレイから得られたデータは有効である。

【0026】図6はSIVTのエントリを表している。図6において、各エントリはタグフィールド303、データフィールド304、バリッドビット305からなる。SIVTはストア命令アドレスのうちエントリ数分に相当する下位ビットでセレクトされる。タグフィールド303には残りの上位ビットが保存されている。セレクトされたエントリ306に保存されているタグとストア命令アドレスの上位ビットを比較し、一致すればそのストア命令アドレスに対応するデータが保持されていることになる。データフィールド304には、該ストア命令がかつて保存したデータが保存されている。バリッドビット305はそのエントリ306が有効であるかどうかを表している。プロセッサの起動時にはバリッドビット305は全て無効を表しており、SIVTの各エント

リ306にデータが保存されるとバリッドビット305が有効になる。

【0027】次に、プロセッサのパイプライン動作と各テーブルの操作との関係を説明する。

【0028】図7に示すパイプライン動作を参照して、各テーブルの参照動作を説明する。

【0029】図7において、IFステージでプログラムカウンタ(PC)600の示すロード命令アドレスを用いてLIST207にアクセスする。LIST207から得られたストア命令アドレスを用いてSIVT307にアクセスする。SIVT307から得られたデータを後続の命令にオペランドとして供給する。ロード命令はデータアドレスの計算を終えずにデータを供給できるので、後続の命令が停止することはない。このように、LIST207とSIVT307から予測されたデータを用いて後続の命令を投機的に実行できる。

【0030】ロード命令のMAステージにおいてデータメモリ800からデータが得られる。このデータとSIVT307から得られたデータとを比較器500で比較する。両者が一致すれば投機実行は成功である。不一致の場合は投機に失敗したので、プロセッサの状態を投機を開始する以前の状態に復帰しなければならない。復帰するためには、リオーダーバッファ等の分岐命令の投機実行に用いられる公知の装置が利用可能である。

【0031】次に、各テーブルへの登録について説明する。

【0032】図8に示すように、ストア命令完了時に常にDIST107とSIVT307に登録が行なわれる。ストア命令の完了時には登録に必要なストア命令アドレス、データアドレス、データは全て揃っているの、登録を行うことができる。ロード命令実行時でLIST207に期待するストア命令アドレスが保持されていなかった場合には、図9に示すようにロード命令完了時にLIST207に登録が行なわれる。ロード命令完了時にはデータアドレスが計算済であるので、これを用いてDIST107にアクセスする。DIST107から得られたストア命令アドレスとロード命令アドレスとを用いてLIST207に登録が行なわれる。

【0033】次に、請求項6又は8記載の発明の一実施形態に係るデータ供給装置について説明する。

【0034】本発明は、標準的なプロセッサを構成する要素と、以下の3つのテーブルとから構成される。すなわち、データアドレスでインデックスされストア命令アドレスを保持するデータインデックスストアテーブル(DIST)、ロード命令アドレスでインデックスされストア命令アドレスを保持するロードインデックスストアテーブル(LIST)、ストア命令アドレスでインデックスされデータの値を保持するストアインデックスバリュートーブル(SIVT)である。各テーブルはキャッシュメモリと同様に構成されており、各テーブルの

連想度には制約はないものとする。なお、ここでは、各テーブルはダイレクトマップ（連想度1）で構成されているものとして説明する。DISTは図1及び図2と同様であり、SIVTは図5及び図6と同様であり、その説明は省略する。

【0035】図10はLISTの動作と構成を表している。図10において、LISTはタグアレイ211とデータアレイ210から構成され、ロード命令アドレスであるエントリ216がセレクトされる。タグアレイ211から得られるタグとロード命令アドレスの上位ビットとを比較し、一致すればデータアレイ210から得られたストア命令アドレスとデータアドレスは有効である。

【0036】図11はLISTのエントリ216を表している。図11において、各エントリ216はタグフィールド213、ストア命令アドレスフィールド214、データアドレスフィールド219、バリッドビット215からなる。LISTはロード命令アドレスのうちエントリ数分に相当する下位ビットでセレクトされる。タグフィールド213には残りの上位ビットが保存されている。セレクトされたエントリ216に保存されているタグとロード命令アドレスの上位ビットを比較し、一致すればそのロード命令アドレスに対応するストア命令アドレスが保持されていることになる。ストア命令アドレスフィールド214には、該ロード命令が参照しようとしているデータを保存したストア命令のアドレスが保存されている。データアドレスフィールド219には該データが保存されているデータアドレスが保存されている。バリッドビット215はそのエントリが有効であるかどうかを表している。プロセッサの起動時にはバリッドビット215は全て無効を表しており、LISTの各エントリにデータが保存されるとバリッドビット215が有効になる。

【0037】次に、パイプライン動作と各テーブルの操作との関係を説明する。

【0038】図12に示すパイプラインの動作を参照して、各テーブルの参照動作を説明する。

【0039】図12において、IFステージでプログラムカウンタ600の示すロードアドレスを用いてLIST217にアクセスする。LIST217から得られたストア命令アドレスを用いてSIVT307にアクセスする。SIVT307から得られたデータを後続の命令にオペランドとして供給する。ロード命令はデータアドレスの計算を終えずにデータを供給できるので、後続の命令が停止することはない。こうして、LIST207とSIVT307から予測されたデータを用いて後続の命令を投機的に実行できる。

【0040】SIVT307に所望のデータが保存されていなかった場合には、LIST207から得られたデータアドレスを用いてデータメモリ800に投機的にアクセスする。データメモリ800が十分高速であれば、

後続の命令を停止させないようにデータを供給することが可能である。ロード命令のMAステージにおいてデータメモリ800からデータが得られると、このデータとSIVTから得られたデータとを比較器500で比較する。もし一致すれば投機実行は成功である。不一致の場合は投機に失敗したので、プロセッサの状態を投機を開始する以前の状態に復帰しなければならない。復帰するためには、リオーダーバッファなど分岐命令の投機実行に用いられる公知の装置が利用可能である。

10 【0041】データメモリ800に投機的にアクセスした場合には、EXステージで実際のデータアドレスが計算されたときに投機の成功判定が行なわれる。投機的アクセスに用いられたデータアドレスと計算されたデータアドレスとを比較器501で比較する。もし一致すれば投機実行は成功である。不一致の場合は投機に失敗したので、プロセッサの状態を投機を開始する以前の状態に復帰しなければならない。復帰するためには上記と同様に、リオーダーバッファなど分岐命令の投機実行に用いられる公知の装置が利用可能である。

20 【0042】次に、各テーブルへの登録について説明する。

【0043】DIST107とSIVT307の登録は上記と同様にして行われる。ストア命令の完了時には登録に必要なストア命令アドレス、データアドレス、データは全て揃っているので登録が行われる。ロード命令実行時でLIST217に期待するストア命令アドレスが保持されていなかった場合には、図13に示すようにロード命令完了時にLIST217に登録が行なわれる。ロード命令完了時にはデータアドレスが計算済であるので、これを用いてDIST107にアクセスする。DIST107から得られたストア命令アドレスと、ロード命令アドレス、データアドレスとを用いてLIST217に登録が行なわれる。

30 【0044】この実施形態は、前述した実施形態に比べて、以下の理由で有利である。SIVT307に所望のデータが登録されていない時、前述した実施形態では予測データが得られないので、投機的実行をすることは不可能である。一方、この実施形態では、予測データアドレス710でデータメモリ800にアクセスすることで投機の実行が可能である。しかし、LIST217にデータアドレスフィールド219が必要であり、その分ハードウェアは増加する。構成を小さくしたい場合には前述した実施形態が有利である。

【0045】次に、請求項7又は8記載の発明の一実施形態に係るデータ供給装置について説明する。

【0046】本発明は、標準的なプロセッサを構成する要素と、以下の3つのテーブルとから構成される。すなわち、データアドレスでインデックスされストア命令アドレスを保持するデータインデックスストアテーブル

50 (DIST)、ロード命令アドレスでインデックスされ

てストア命令アドレスを保持するロードインデックスストアテーブル (LIST)、ストア命令アドレスでインデックスされデータの値を保持するストアインデックスバリュートーブル (SIVT) である。各テーブルはキャッシュメモリと同様に構成されており、各テーブルの連想度には制約はないものとする。なお、ここでは、各テーブルはダイレクトマップ (連想度 1) で構成されているものとして説明する。DIST は図 1 及び図 2 と同様であり、LIST は図 3 及び図 4 と同様であり、その説明は省略する。

【0047】図 14 は SIVT の動作と構成を表している。図 14 において、SIVT はタグアレイ 311 とデータアレイ 310 とから構成され、ストア命令アドレスであるエントリ 316 がセレクトされる。タグアレイ 311 から得られるタグとストア命令アドレスの上位ビットとを比較し、一致すればデータアレイ 311 から得られたデータは有効である。

【0048】図 15 は SIVT のエントリを表している。図 15 において、各エントリはタグフィールド 313、データフィールド 319、バリッドビット 315 からなる。SIVT はストア命令アドレスのうちエントリ数分に相当する下位ビットでセレクトされる。タグフィールド 313 には残りの上位ビットが保存されている。セレクトされたエントリに保存されているタグとストア命令アドレスの上位ビットを比較し、一致すればそのストア命令アドレスに対応するデータとデータアドレスが保持されていることになる。データフィールド 314 には、該ストア命令がかつて保存したデータが保存されている。データアドレスフィールド 319 はそのデータが保持されているアドレスが保存されている。バリッドビット 315 はそのエントリ 316 が有効であるかどうかを表している。プロセッサの起動時にはバリッドビット 315 は全て無効を表しており、SIVT の各エントリ 316 にデータが保存されるバリッドビット 315 が有効になる。

【0049】次に、パイプライン動作と各テーブルの操作との関係を説明する。

【0050】図 16 に示すパイプラインの動作を参照して、各テーブルの参照動作を説明する。

【0051】図 16 において、IF ステージでプログラムカウンタ 600 の示すロード命令アドレスを用いて LIST 207 にアクセスする。LIST 207 から得られたストア命令アドレスを用いて SIVT 317 にアクセスする。SIVT 317 から得られたデータを後続の命令にオペランドとして供給する。ロード命令はデータアドレスの計算を終えずにデータを供給できるので、後続の命令が停止することはない。こうして、LIST 207 と SIVT 317 から予測されたデータを用いて後続の命令を投機的に実行できる。

【0052】ロード命令の EX ステージにおいてデータ

アドレスが計算されると、このデータアドレスと SIVT 317 から得られたデータアドレスとを比較器 501 で比較する。もし一致すれば投機実行は成功である。不一致の場合は投機に失敗したので、プロセッサの状態を投機を開始する以前の状態に復帰しなければならない。復帰するためには、リオーダーバッファなど分岐命令の投機実行に用いられる公知の装置が利用可能である。

【0053】各テーブルへの登録は、上述した実施形態と同様にして行われる。

10 【0054】この実施形態は、図 1～図 9 に示す実施形態と比較して以下の理由で有利である。先の実施形態では投機の成功判定はロード命令の MA ステージにならなければならない。一方、この実施形態は EX ステージで可能である。投機に成功した場合には両者に差はないが、失敗した場合にはこの実施形態の方が回復を早期に行なえる分性能上有利である。しかし、SIVT 317 にデータアドレスフィールド 319 が必要であり、その分ハードウェアは増加する。したがって、構成を小さくしたい場合には先の実施形態が有利である。

20 【0055】次に、請求項 7 または 8 記載の発明の一実施形態に係るデータ供給装置について説明する。

【0056】本発明は、標準的なプロセッサを構成する要素と、以下の 3 つのテーブルとから構成される。すなわち、データアドレスでインデックスされストア命令アドレスを保持するデータインデックスストアテーブル (DIST)、ロード命令アドレスでインデックスされストア命令アドレスを保持するロードインデックスストアテーブル (LIST)、ストア命令アドレスでインデックスされデータの値を保持するストアインデックスバリュートーブル (SIVT) である。各テーブルはキャッシュメモリと同様に構成されており、各テーブルの連想度には制約はないものとする。なお、ここでは、各テーブルはダイレクトマップ (連想度 1) で構成されているものとして説明する。DIST は図 1 及び図 2 と同様であり、LIST は図 10 及び図 11 と同様であり、SIVT は図 14 及び図 15 と同様であり、その説明は省略する。

【0057】続いて、パイプライン動作と各テーブルの操作との関係を説明する。

40 【0058】図 17 に示すパイプラインの動作を参照して、各テーブルの参照動作を説明する。

【0059】図 17 において、IF ステージでプログラムカウンタ 600 の示すロードアドレスを用いて LIST 217 にアクセスする。LIST 217 から得られたストア命令アドレスを用いて SIVT 317 にアクセスする。SIVT 317 から得られたデータを後続の命令にオペランドとして供給する。ロード命令はデータアドレスの計算を終えずにデータを供給できるので、後続の命令が停止することはない。こうして、LIST 217 と SIVT 317 から予測されたデータを用いて後続の

命令を投機的に実行できる。

【0060】SIVT317に所望のデータが保持されていなかった場合には、LIST217から得られたデータアドレスを用いデータメモリ800に投機的にアクセスする。データメモリ800が十分高速であれば、後続の命令を停止させないようにデータを供給することが可能である。ロード命令のEXステージにおいてデータアドレスが計算されると、このデータアドレスとSIVT317から得られたデータアドレスとを比較器501で比較する。もし一致すれば投機実行は成功である。不一致の場合は投機に失敗したので、プロセッサの状態を投機を開始する以前の状態に復帰しなければならない。復帰するためには、リオーダバッファなど分岐命令の投機実行に用いられる公知の装置が利用可能である。マルチプレクサ502は、SIVT317に所望のデータが保持されていない場合はLIST217のデータアドレスを選択し、SIVT317に所望のデータが保持されている場合にはSIVT307のデータアドレスを選択する。

【0061】ロード命令のEXステージにおいてデータメモリ800からデータが得られると、このデータとSIVT307から得られたデータアドレスとを比較する。もし一致すれば投機実行は成功である。不一致の場合は投機に失敗したので、プロセッサの状態を投機を開始する以前の状態に復帰しなければならない。復帰するためには、リオーダバッファなど分岐命令の投機実行に用いられる公知の装置が利用可能である。

【0062】各テーブルへの登録は、前述した実施形態と同様にして行われる。

【0063】このような実施形態においては、図10～図16に示す実施形態で得られる効果を得ることができる。

【0064】次に、請求項9記載の発明の一実施形態について説明する。

【0065】この実施形態は、上述した実施形態に対して、LISTにアクセスするステージと、SIVTにアクセスするステージを異なるステージしたことを特徴とする。

【0066】IFステージでLISTにアクセスする。IDステージでLISTから得られたストア命令アドレスを用いてSIVTにアクセスする。SIVTから得られたデータを、後続の命令にオペランドとして供給する。ロード命令はデータアドレスの計算を終えずにデータを供給できるので、後続の命令が停止することはない。こうして、LISTとSIVTから予測されたデータを用いて後続の命令を投機的に実行できる。

【0067】LISTおよびSIVTが低速であった場合には、これら2つのテーブルをIFステージでアクセスすることで、プロセッサのサイクルタイムを伸ばしてしまうおそれがある。しかし、異なるステージでアクセ

スすれば上記の問題はなくなる。なぜなら、これらのテーブルはキャッシュメモリと同様な構成であり、キャッシュメモリよりも低速にはならないのであるから、サイクルタイムを伸ばすこともあり得ない。逆に、LISTおよびSIVTが高速な場合に例えばそれらの容量が小さい場合には、前述した実施形態の構成を取ることが可能である。

【0068】次に、請求項4記載の発明の一実施形態について説明する。

10 【0069】この実施形態は、前述した実施形態において、SIVTの各エントリにあるバリッドビットを、図18に示すようにバイト単位に設けるようにしたことを特徴とする。こうすることで、ストア命令の扱うデータ幅がロード命令の扱うデータ幅よりも小さい場合に対処することができる。ロード命令のデータ幅に比べてストア命令のデータ幅が小さい場合には、ロード命令が参照するバリッドビットが無効になっているので、確実に失敗する投機を行なわないで済む。

20 【0070】次に、請求項5記載の発明の一実施形態について説明する。

【0071】この実施形態は、前述した実施形態において、図19に示すように、LISTの各エントリに予測の成功を数えるカウンタフィールド228を設けたことを特徴とする。カウンタフィールド228のビット幅には特に制限はないので、ここでは2ビットの例を説明する。予測が成功した時はカウンタフィールド228の値を+1増加させ、失敗した時にはカウンタフィールド228の値を-1減少させる。LISTを参照した時に、カウンタフィールド228の値が例えば2以上であれば投機的実行を行ない、そうでなければ投機を行なわない。この実施形態は、各ロード命令に対して投機的実行の有効性をカウンタフィールド228に登録することで、不用意な投機的実行を抑制し、投機の失敗による性能低下を防ぐことが可能である。

【0072】

【発明の効果】以上説明したように、請求項1、2又は3記載の発明によれば、DIST、LISTならびにSIVTのテーブルを関連づけて機能させることによりロード命令の命令アドレスからデータ値を予測するようにしたので、ロード命令がデータアドレス計算を終了する前にデータを獲得できるので、ロード命令の実行を早くでき、さらに予測されたデータを用いて後続の命令を投機的に実行することによりプロセッサの性能を向上できる。一方、ストア命令とロード命令の関係をデータアドレスの介在なしに検出できるため、ストア命令とロード命令の同時実行や、ロード命令によるストア命令の追い越し実行が可能になり、プロセッサの性能を向上できる。

50 【0073】請求項4記載の発明によれば、ストア命令の扱うデータ幅がロード命令の扱うデータ幅よりも小さ

い場合に対処することができる。

【0074】請求項5記載の発明によれば、不用意な投機的な実行を抑制し、投機の失敗による性能低下を防ぐことができる。

【0075】請求項6記載の発明によれば、予測データアドレスがより速く得られるために投機的実行を迅速に開始することができる。

【0076】請求項7又は8記載の発明によれば、予測データによる投機的な実行の成否が速く判別できるため、投機に失敗した時の回復を早期に行うことができる。

【0077】請求項9記載の発明によれば、プロセッサのサイクルタイムが長くなることを回避できる。

【図面の簡単な説明】

【図1】請求項1、2又は3記載の発明の一実施形態に係るプロセッサのデータ供給装置におけるDISTの構成を示す図である。

【図2】DISTのエントリを示す図である。

【図3】請求項1、2又は3記載の発明の一実施形態に係るプロセッサのデータ供給装置におけるLISTの構成を示す図である。

【図4】LISTのエントリを示す図である。

【図5】請求項1、2又は3記載の発明の一実施形態に係るプロセッサのデータ供給装置におけるSIVTの構成を表す図。

【図6】SIVTのエントリを示す図である。

【図7】ロードデータの予測を示す図である。

【図8】DISTおよびSIVTへの登録を示す図である。

【図9】LISTへの登録を示す図である。

【図10】請求項6記載の発明の一実施形態に係るプロセッサのデータ供給装置におけるLISTの構成を示す図である。

【図11】LISTのエントリを示す図である。

【図12】ロードデータの予測を示す図である。

【図13】LISTへの登録を示す図である。

【図14】請求項7記載の発明の一実施形態に係るプロセッサのデータ供給装置におけるSIVTの構成を示す図である。

【図15】SIVTのエントリを示す図である。

【図16】ロードデータの予測を示す図である。

【図17】ロードデータの予測を示す図である。

【図18】SIVTのエントリを示す図である。

【図19】LISTのエントリを示す図である。

【図20】パイプライン処理における5段パイプラインを示す図である。

【図21】パイプライン処理におけるデータハザードを示す図である。

【図22】潜在的なアドレス衝突によるハザードを示す図である。

【符号の説明】

- 100 DISTのデータアレイ
- 101 DISTのタグアレイ
- 102, 202, 302, 312, 500, 501, 502 比較器
- 103 DISTエントリのタグフィールド
- 104 DISTエントリのストア命令アドレスフィールド
- 105 DISTのエントリのバリッドビット
- 106 DISTエントリ
- 200, 210 LISTのデータアレイ
- 201, 211 LISTのタグアレイ
- 203, 213 LISTエントリのタグフィールド
- 204, 214 LISTエントリのストア命令アドレスフィールド
- 205, 215 LISTのエントリのバリッドビット
- 206, 216 LISTエントリ
- 207, 217 LIST
- 208 LISTの供給するストア命令アドレス
- 219 LISTエントリのデータアドレスフィールド
- 300 SIVTのデータアレイ
- 301 SIVTのタグアレイ
- 303, 313 SIVTエントリのタグフィールド
- 304, 314 SIVTエントリのデータ値フィールド
- 305, 315 SIVTエントリのバリッドビット
- 306, 316 SIVTエントリ
- 307, 317 SIVT
- 310 SIVTのデータアレイ
- 311 SIVTのタグアレイ
- 319 SIVTエントリのデータアドレスフィールド
- 400 600 PC
- 800 データメモリ

【図2】

103	104	105
タグ	ストアアドレス	バリッド

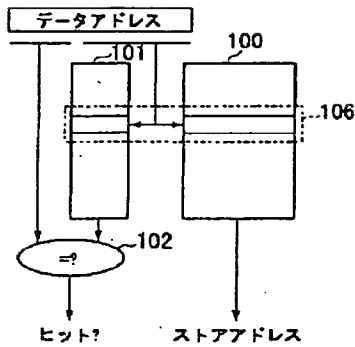
【図4】

203	204	205
タグ	ストアアドレス	バリッド

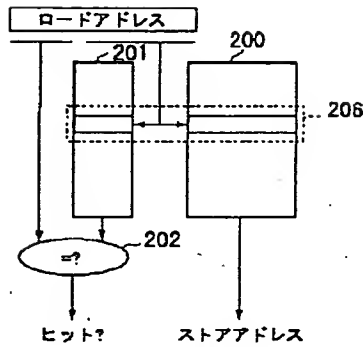
【図6】

303	304	305
タグ	データ	バリッド

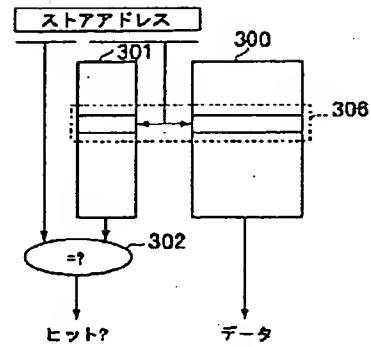
【図1】



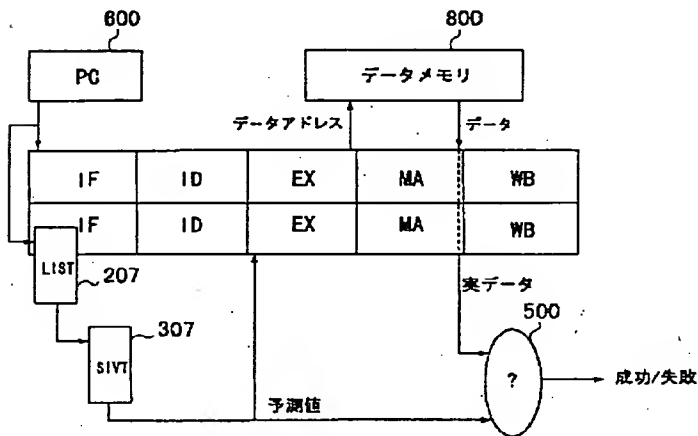
【図3】



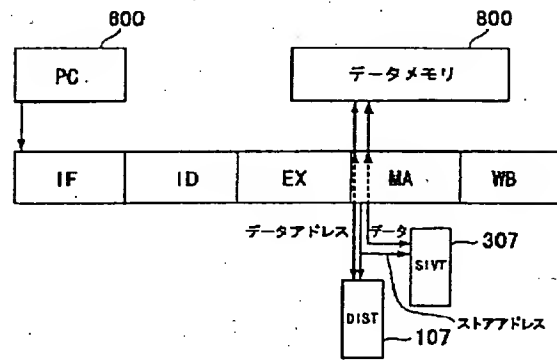
【図5】



【図7】

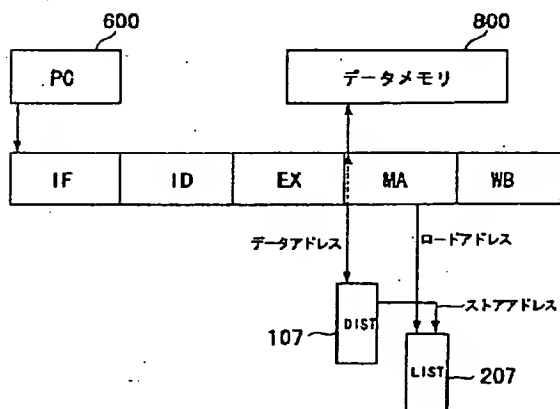


【図8】

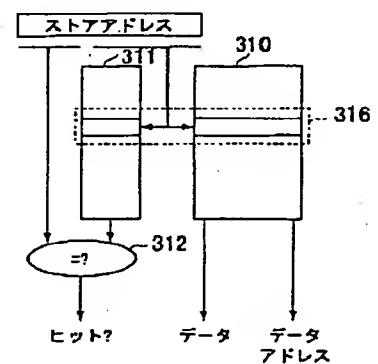
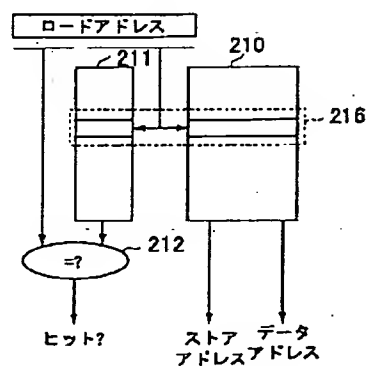


【図14】

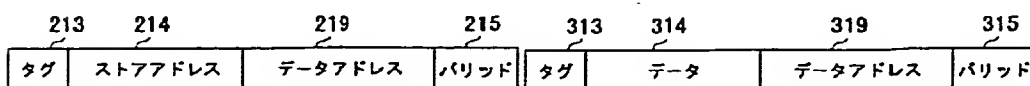
【図9】



【図10】

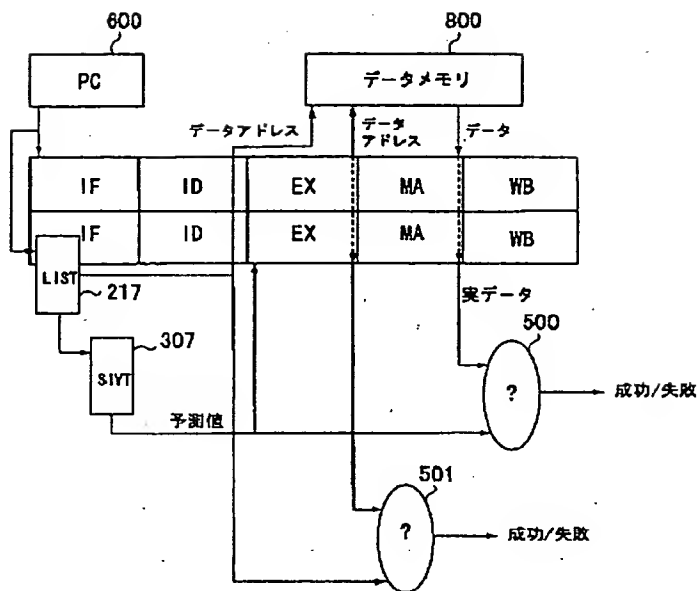


【図11】

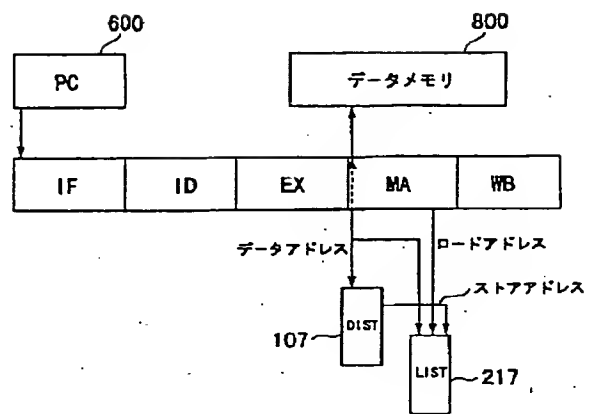


【図15】

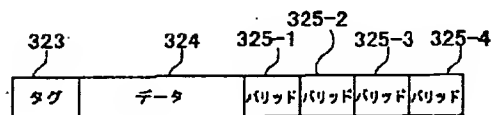
【図 1 2】



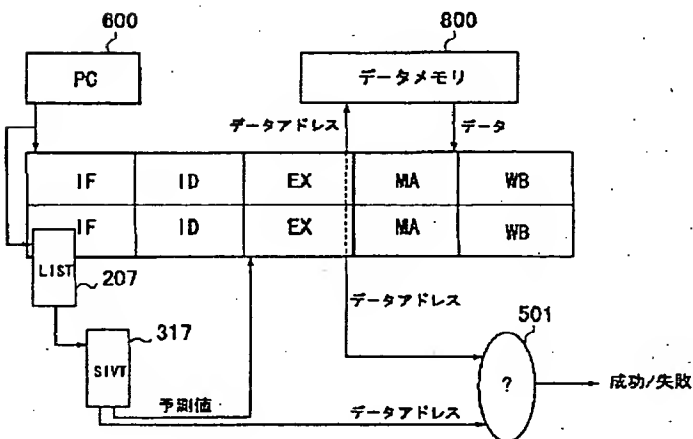
【図 1 3】



【図 1 8】

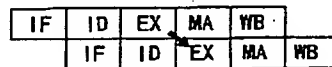


【図 1 6】



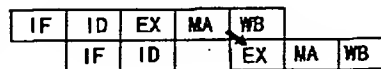
【図 2 1】

add r1 ← r2 + r5
add r3 ← r1 + r20

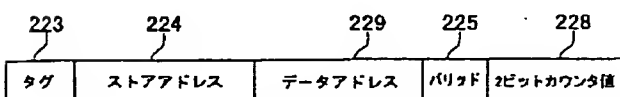


lw r1 ← r2(0)

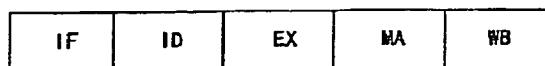
add r3 ← r1 + r20



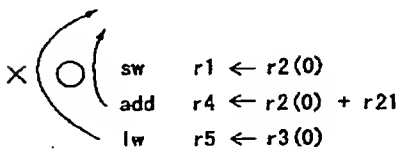
【図 1 9】



【図 2 0】



【図 2 2】



【図 17】

